Acta Academiae Paedagogicae Agriensis, Sectio Mathematicae, 25. (1998) pp. 103–107

Interpolation possibilities using rational B-spline curve

MIKLÓS HOFFMANN and EMŐD KOVÁCS

Abstract. The aim of this paper is to solve some interesting interpolation problems using rational B-spline curve. If a sequence of planar points and vectors are given then a free-form curve can calculated which interpolates the points and has the given tangent vectors in these points. Our method gives a fast interpolation of these data using extra control points. Then we provide a method which allows to interpolate the same set of data without any predefined order of the points, i.e. a set of scattered points with the vectors. In this latter problem we use an artificial neural network to order the data.

Introduction

Rational B-spline curves and surfaces (or simply called NURBS), as the generalization of B-spline curve and surface, are widely used in CAD/CAM, and free-from design [4]. Basicly these methods have been developed for approximating points, but they can be used as interpolating curves or surfaces as well. In this paper we will use the rational B-spline curve for a special interpolation problem, where beside the points the tangent vectors of the future curve are also given. The method is similar to the case of B-spline: the control points of the future curve is calculated from the given data, so finally it will be an approximating curve, but given points will be on the curve and it will have the given tangent vectors.

This problem can also be fomulated without giving the order of points. Since all the basic free-form methods are defined with a sequence of points as input data, in this case we use an artificial neural network, the Kohonen net, to order the points and then we apply the method mentioned above.

Interpolation of a sequence of points and vectors

At first we define the rational B-spline curve as an approximating curve. If a sequence of points $V_i, i = 1, \dots, n$ (called control points) and positive real numbers $w_i, i = 1, \dots, n$ (called weights) are given, then the third order

This research was supported by the Hungarian National Foundation for Scientific Research (OTKA), grant No. F019395.

uniform rational B-spline curve can be defined as follows:

$$Q_i(u) = \frac{\sum_{r=-1}^{2} V_{i+r} w_{i+r} b_r(u)}{\sum_{r=-1}^{2} w_{i+r} b_r(u)} \qquad u \in [0,1]$$

where the functions b_r are the well-known basic functions:

$$\begin{aligned} b_{-1}(u) &= \frac{1 - 3u + 3u^2 - u^3}{6} \\ b_0(u) &= \frac{4 - 6u^2 + 3u^3}{6} \\ b_1(u) &= \frac{1 + 3u + 3u^2 - 3u^3}{6} \\ b_2(u) &= \frac{u^3}{6} \end{aligned}$$

Note, that the curve consists of segments, and the parameter u runs over the interval $[0,1]$ in every segment. This fact will be strongly used in the basic idea of the interpolation.

Now let a sequence of points $P_i, i = 1, \ldots, m$ and a sequence of vectors $t_i, i = 1, \ldots, m$ be given. Find a sequence of contol points V_j and weights w_j (at this moment the number of points and weights is unknown) such that the curve using these control points and weights interpolates the points P_i and has the tangent vectors t_i in these points.

Suppose, that the points P_i will be the starting points of the segments of the future curve. Considering the properties of the segments mentioned above, this assumption can be formulated as follows:

$$\begin{aligned} Q_i(0) &= P_i \qquad i = 1, \ldots, m-1 \\ Q_m(1) &= P_m \end{aligned}$$

The last equation means that the last point P_m would be the end point of the last segment.

On the other hand, the given vectors have to be equal to the derivatives of the curve in the starting points of the segments:

$$\begin{aligned} \dot{Q}_i(0) &= t_i \qquad i = 1, \ldots, m-1 \\ \dot{Q}_m(1) &= t_m \end{aligned}$$

Unfortunately these equations yield a second order system of equations for the control points V_j and the weights w_j. To reduce the complication and save computing time, some control points will be defined in advance with unit weight. Let these points be the intersections of the line of the tangent vectors t_i, t_{i+1} or, if this point would be too far from the points P_i, simply be the midpoint of the section P_i, P_{i+1}. The position of these contol points are not important because later on they can be modified without disturbing the interpolation.

Hence the system of equations has to be solved is the following:

$$\frac{\frac{1}{6}w_{i-1}V_{i-1} + \frac{4}{6}w_iV_i + \frac{1}{6}w_{i+1}V_{i+1}}{\frac{1}{6}w_{i-1} + \frac{4}{6}w_i + \frac{1}{6}w_{i+1}} = P_i$$

$$\frac{(-\frac{1}{2}w_{i-1}V_{i-1} + \frac{1}{2}w_{i+1}V_{i+1})(\frac{1}{6}w_{i-1} + \frac{4}{6}w_i + \frac{1}{6}w_{i+1})}{(\frac{1}{6}w_{i-1} + \frac{4}{6}w_i + \frac{1}{6}w_{i+1})^2} -$$

$$-\frac{(\frac{1}{6}w_{i-1}V_{i-1} + \frac{1}{6}w_iV_i + \frac{1}{6}w_{i+1}V_{i+1})(-\frac{1}{2}w_{i-1} + \frac{1}{2}w_{i+1})}{(\frac{1}{6}w_{i-1} + \frac{4}{6}w_i + \frac{1}{6}w_{i+1})^2} = t_i$$

In these equations only V_i and w_i are unknown, since $V_{i-1}, V_{i+1}, w_{i-1}, w_{i+1}$ are predefined. Coefficients can be calculated by the functions $b_i(u)$ and $\dot{b}_i(u)$ in $u = 0$.

Now the solutions of these equations and the predefined control points and weights form a control poligon and a sequence of weights, with which the rational B-spline curve can be drawn, and it will pass through the points P_i and will have the tangent vectors t_i.

Interpolation of scattered points with tangent vectors

This problem is similar to the previous one, but the given points have no predefined order, i.e. we do not know which point has to be the first and which one is the last one. Since the rational B-spline method can be applied only on a sequence of points (and weights), first of all we have to order the points. For this purpose an artificial neural network will be used. After this step the same procedure described above can be applied. Now after a short description of the applied net, the Kohonen network [2], the ordering process and the interpolation will be discussed. For more detailed discussion of the ordering method by Kohonen network see [1]–[2], [5].

The Kohonen neural network is a two-layered non-supervised learning neural network. Self organizing networks, like the applied Kohonen net, organize the input data during the so called learning phase without any supervision. The most important part of the algorithm is the training rule, which modifies the network according ot the input points.

Let a set of points $P_i (i = 1, \ldots, n)$ (scattered data) and a set of vectors $t_i (i = 1, \ldots, n)$ be given on the plane. Our first task is to determine the order of the points for the interpolation problem.

The Kohonen net is used to order the points. The first layer of neurons is called input layer and contains the two input neurons which pick up the data, the planar points. The input neurons are entirely interconnected to a second, competitive layer, containing m neurons (where $m \geq n$, usually $m = 4n$). The weights associated with the connections are adjusted during training by the following rule:

— Coordinates of the scattered points: $P_i(x_{1i}, x_{2i}, x_{3i}) \quad (i = 1, \dots, n)$

— Coordinates of the output points: $Q_j(w_{1j}, w_{2j}, w_{3j}) \quad (j = 1, \dots, m)$

STEP 1. Initialize the weights $w_{sj}, \quad (s = 1, 2, 3 \quad j = 1, \dots, m)$ as small random values around the average of the coordinates of the input points. Let the training time $t = 1$

STEP 2. Present new input values $(x_{1i_0}, x_{2i_0}, x_{3i_0})$, as the coordinates of a randomly selected input point p_{i_0}

STEP 3. Compute the Euclidean distance of all output nodes to the input point:

$$d_j = \sum_{s=1}^{3} (x_{si_0} - w_{sj})^2$$

STEP 4. Find the winning unit q_{j_0} as the node which has the minimum distance to the input point, so where j_0 is the value for which $d_{j_0} = min(d_j)$

STEP 5. Compute the neighborhood $N(t) = (j_0, j_1, \dots, j_k)$

STEP 6. Update the weights (i.e.the coordinates) of the nodes in the neighborhood by the following equation:

$$w_{sj}(t+1) = w_{sj}(t) + \eta(t)(x_{si_0} - w_{sj}(t)) \quad \forall j \in N(t)$$

where $\eta(t)$ is a so called gain term, a Gaussian function decreasing in time.

STEP 7. Let $t = t + 1$. Repeat STEP 2-7 until the network is trained.

The network is said to be trained if all the input points are on the polygon, that is for all the input points $P_i(i = 1, \dots, m)$ there is an output vector o_j such that after a certain time t_0 the Euclidean distance of o_j and P_i is smaller thane a predefined limit. A stronger convergence can be obtained if we require that the output vectors which do not converge to an input vector be on the line determined by its two neighbouring output vectors. This stronger convergence is important especially in term of the smoothness of the future curve. For the detailed description and evaluation of this problem see [2].

After the ordering process the same algorithm can be applied to calculate the interpolation curve as we described above. At this part of the process it is irrelevant, that the input points were scattered.

Conclusions

In the free-form design there are several different method according to the problem (approximation or interpolation) and the type of data (ordered or scattered). In this paper we provided two algorithms, with the help of which all kinds of problems and types of data can be handled by the rational B-spline curve. Even if parts of the data are scattered and others have to be interplated or approximated, the final result (joining the calculated control polygons) will be a unique curve.

References

[1] M. Hoffmann, L. Várady, Free-form curve design by neural networks, *Acta. Acad. Paed. Agriensis,* Tom. XXIV., 1997, 99–104.

[2] M. Hoffmann, L. Várady, Free-form surface design by neural networks, *Journal for Geometry and Graphics,* Vol. 2. N.1., 1998., 1–6.

[3] T. Kohonen: *Self-organization and associative memory,* Springer-Verlag, 1984.

[4] L. Piegl, W. Tiller: *The NURBS Book,* Springer Verlag, 1997.

[5] L. Várady: Analysis of the Dynamic Kohonen Network Used for Approximating Scattered Data, *Proceedings of the 7th ICECGDG,* Cracow, 1996, 433–436.

Miklós Hoffmann and Emőd Kovács
Institute of Mathematics and Computer Science
Károly Eszterházy Teachers' Training College
Leányka u. 4.
3300 Eger, Hungary
e-mail: hofi@ektf.hu, emod@ektf.hu